

TITLE OF THE INVENTION

IMAGE PROCESSING METHOD AND APPARATUS

5

FIELD OF THE INVENTION

The present invention relates to image processing method and apparatus for detecting the direction (upward/downward/leftward/rightward) of an original 10 image inputted using a scanner or the like.

BACKGROUND OF THE INVENTION

Conventionally, the direction of an original image inputted into a computer using a scanner or the 15 like is detected by the following methods:

(1) Detection of original image direction by software

Fig. 30 shows the outline of software processing for detecting the direction of an image. As shown in 20 Fig. 30, first, a color image 1011 as the object of direction detection is binarized by a binarization processing procedure 1012, and a binary image 1013 is generated. Next, the binary image 1013 is area-divided by an area division processing procedure 1014, and 25 character coordinate information 1015 as coordinate information of a character area is generated. Then, the binary image 1013 in the character coordinate

information 1015 is referred to by an OCR processing procedure 1016 thus character recognition processing is performed, and the result of direction detection of the color image 1011 is outputted.

The above processing will be described in more detail. Fig. 31 shows an example of the respectively 8-bit RGB color image 1011. This color image is simply binarized with a fixed threshold value 128 by luminance conversion. Fig. 32 shows an example of the binary image simply binarized from the color image in Fig. 31. As the binary image in Fig. 32 obtained by simple binarization includes a large amount of noise, binarization processing cannot be performed.

Accordingly, in the binarization processing procedure 1012, a histogram of luminance information of the color image is created as shown in Fig. 33, and an optimum binarization point 1041 is calculated. Fig. 33 shows the luminance information of the color image in Fig. 31 and the optimum binarization point 1041.

Further, Fig. 34 shows a binary image from the color image 1011 using the binarization point 1041. As shown in Fig. 34, as a threshold value using the histogram of luminance information (binarization point 1041) is used in place of the fixed threshold value, the amount of noise included in the binary image in Fig. 34 is smaller in comparison with the binary image in Fig. 32, and excellent area division can be performed.

Fig. 35 shows an example of the result of area division performed on the binary image in Fig. 34 by the area division processing procedure 1014. In the area division processing, the resolution is reduced so 5 as to connect black pixels, then edge line tracing is performed, and it is determined whether or not the traced profile is a character from the shape of the profile. In Fig. 35, rectangular areas 1061 to 1068 are determined as character areas. Note that areas 10 1067 and 1068 are erroneously determined areas.

As described above, in the OCR processing procedure 1016, the areas determined as the character areas are read and character-cutting processing is performed, and direction detection processing is 15 performed on each character. In the direction detection processing, a feature vector of 1 character is calculated, and the feature vector is rotated and character recognition processing in 4 (0° , 90° , 180° and 270°) directions. From the results of character 20 recognition in the 4 directions, one direction with the highest accuracy is determined as the result of direction detection.

In image direction detection by software processing, a value obtained by addition of OCR results 25 of all the characters existing the original (characters existing in the character areas resulted from the area division), is outputted as the final result.

(2) Detection of original image direction by hardware

Next, the outline of hardware construction for the conventional image direction detection processing will be described. Fig. 36 is a block diagram showing the hardware construction of a specialized direction detection board connected to a main board of a monochrome digital copier. In Fig. 36, numeral 1071 denotes a character extraction unit having a specialized GA for character extraction processing and binarization processing; 1072, a RAM; 1073, a CPU; and 1074, a ROM.

Fig. 37 is a timing chart showing respective operations using the direction detection board in Fig. 36. Next, the operations of the direction detection board in Fig. 36 will be described with reference to the timing chart of Fig. 37.

In Fig. 37, "0" to "3" are page numbers of original images placed on an ADF (Automatic Direction Finder) of the monochrome digital copier. Further, numeral 1081 indicates timing of original reading by a scanner of the copier; 1082, timing of the character extraction processing and the binarization processing by the specialized character extraction GA; 1083, timing of direction determination OCR by the CPU; and 1084, timing of outputting the result of direction determination. As shown in Fig. 37, the results of

processing of the respective pages are outputted by 2-pipeline delay from input of original.

First, the scanner sequentially reads the originals placed on the ADF, and VIDEO in Fig. 36 is 5 generated. Note that VIDEO includes a CLK and image data (8 bits) synchronized with the CLK, a page signal indicating a page break of the image data, and a main-scanning synchronizing signal indicating a break of width of the image data.

10 The character extraction unit 1071 inputs the image data (8 bits), detects an area likely to be a character area including continuous image data (more specifically, refers to adjacent plural pixels, and detects an area where the difference between a maximum 15 and minimum value is greater than a threshold value) and generates coordinate data thereof. Further, the character extraction unit 1071 binarizes the image data (8 bits). Note that a threshold value used in the binarization is determined from a histogram of a previous line. Then, the coordinate data and the 20 binary image are written into the RAM 1072 (Note that the GA may have a RAM for storing the data). The above operation is performed at the timing indicated with the numeral 1082 in Fig. 37.

25 Next, the CPU 1073 performs the direction determination OCR processing in parallel to the character extraction processing on the next page. The

CPU 1073 reads the coordinate data from the RAM 1072 in accordance with a program stored in the ROM 1074, and performs the direction determination OCR processing on the binary image on the RAM 1072 corresponding to the coordinate data. Note that in a case where the RAM has a sufficient size, the program on the ROM 1074 is downloaded to the RAM 1072 for increasing the processing speed. The character extraction in synchronization with the CLK is fixed time processing, however, as the direction determination OCR is time-variable processing, the OCR processing is forcibly terminated by utilizing a timer. The results of direction detection within a limited period by the timer (0, 90, 180, 270 and UNKNOWN) are outputted at timing indicated with "Δ" at the timing 1084.

However, the above-described conventional methods have the following problems.

(1) Problems in detection of original image direction by software

(1-1) Much processing time is required.

Hereinbelow, described is a result of measurement of an A4 size image processed by a personal computer having a 266 MHz Pentium (registered trademark) II. First, it takes 1.8 seconds to create a histogram and an calculate an optimum binarization point. Next, it takes 0.3 to 1.0 seconds to perform the area division processing, although the time of the processing varies

in accordance with image (depending on the number of connected black pixels). Then it takes 2 to 3 seconds to perform the OCR processing on the document original mainly including characters, although the time of the 5 processing varies in accordance with the number of characters. Accordingly, total 4 to 5 seconds are required.

(1-2) A large amount of work memory is required.

As the entire color image is referred to so as to 10 obtain an optimum binary image, in case of A4 size image, a 24 MBytes memory is required.

(2) Problems in detection of original image direction by hardware

15 (2-1) Cost is high.

As the specialized board is utilized and the CPU, the RAM, the ROM, the character extraction GA, a control GA (not shown) and the like are necessary only for the direction determination processing, the cost is 20 high.

(2-2) Version updating is difficult

As the character extraction unit is comprised of a specialized GA, version updating of character extraction algorithm cannot be made without difficulty.

25

(3) Problem common to both detection methods

In both methods, it is impossible to perform the

OCR processing on an inverted character portion. In recent years, color office documents are widely used as well as printed documents, and the color images often include more inverted character portions in comparison 5 with monochrome originals. Accordingly, in both methods, the accuracy of recognition is low in color images having inverted character portions.

SUMMARY OF THE INVENTION

10

The present invention has been proposed to solve the conventional problems, and has its object to provide image processing method and apparatus which efficiently detect an image input direction of images, 15 from an image having much differential information to an image having little differential information.

According to the present invention, the foregoing object is attained by providing an image processing method for detecting a direction of an image including 20 a character area, inputted into a computer, the method comprising: a binary image generation step of generating a binary image of the image; a tile image generation step of generating a tile image by applying a predetermined value to tiles, each corresponding to a 25 predetermined size area in the binary image; a character area extraction step of extracting an area in the binary image, corresponding to an area in a

circumscribed rectangle surrounding connected pixels having the same value in the tile image, as a character area; and a direction detection step of recognizing a direction of characters included in the character area
5 and thereby detecting the direction of the image.

Further, in the image processing method, at the binary image generation step, the binary image is generated with image area flags having a value 1 corresponding to a pixel equal to or greater than a
10 predetermined value or a value 0 corresponding to a pixel less than the predetermined value, and at the tile image generation step, the tile image is generated with a tile having a value 1 where the number of image area flags having the value 1 is equal to or greater
15 than a predetermined threshold value, and a tile having a value 0 where the number of image area flags having the value 1 is less than the predetermined threshold value.

Further, the image processing method, further
20 comprises: a character extraction step of extracting the respective characters included in the character area extracted at the character area extraction step; and a character recognition step of recognizing a direction of the characters extracted at the character
25 extraction step, and at the direction detection step, the direction of the character area is detected based on the result of recognition of the direction of the

characters included in the character area.

Further, the image processing method further comprises: a determination step of determining whether or not the character area is an inverted image based on 5 the binary image of the image; and an inversion processing step of inverting black and white components of the binary image if it is determined at the determination step that the character area is an inverted image.

10 Further, in the image processing method, at the tile image generation step, plural tile images are generated using plural different threshold values, and at the character area extraction step, the plural tile images are compared with each other and the character 15 area included in the image is extracted.

Further, in the image processing method, the tile image is a low resolution binary image generated by counting the binary image, generated by differentiating the image, by a small area.

20 Further, in the image processing method, the tile image is a low resolution differential image generated by counting differential information of the image by a small area.

Further, in the image processing method, at the 25 character area detection step, an area in the image, corresponding to the connected pixels extracted from the low resolution image, is extracted as a character

area.

Further, in the image processing method, at the tile image generation step, plural low resolution images are generated using plural different threshold values.

Further, in the image processing method, at the character area extraction step, connected pixels extracted from the plural low resolution images are compared with the plural low resolution images and the character area is extracted.

Further, in the image processing method, at the character area extraction step, the low resolution image is divided into meshes, and the character area is extracted based on distribution of pixels within each mesh area.

Further, in the image processing method, the character area extraction step includes a selection output step of selectively outputting a character area extracted using connected pixels extracted from the low resolution image and a character area determined based on the distribution of pixels within each mesh area.

According to the present invention, as a character area can be detected utilizing plural low-resolution images comprised of differential information of a color image, an image input direction can be efficiently detected in images, from an image with much differential information to an image with little

differential information.

Further, according to the present invention, as an inversion determination unit is provided, character recognition can be performed on an inverted character portion. Accordingly, even in a color image having many inverted character portions, the input direction can be detected.

Further, according to the present invention, as high-speed software processing can be performed with a small amount of work memory, the cost of parts upon version update of direction detection processing is unnecessary.

Other features and advantages of the present invention will be apparent from the following description taken in conjunction with the accompanying drawings, in which like reference characters designate the same name or similar parts throughout the figures thereof.

20

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate embodiments of the invention and, together with the description, serve to explain the principles of the invention.

Fig. 1 is a block diagram showing the

construction of an image processing apparatus according to a first embodiment of the present invention;

Fig. 2 is a block diagram showing a hardware construction necessary for direction detection
5 processing in a case where a digital color copier in Figs. 29A to 29C is applied to a general image processing system 102;

Fig. 3 is a timing chart showing respective operations of an image processing system 202 in Fig. 2;
10 Fig. 4 is an example of a color image used in the direction determination processing according to the first embodiment;

Fig. 5 is an example of an image as a result of tile header generation with threshold values $T_1 = 15$ and $T_2 = 1$ in a scanner image processor 2114;

Fig. 6 is a graph showing a whole histogram of the color image in Fig. 4;

Fig. 7 is an example of a processed result showing 3 rectangular areas finally extracted by
20 primary character extraction processing;

Fig. 8 is an example of a read image from an area 701;

Fig. 9 is an example of a read image from an area 702;

25 Fig. 10 is an example of a read image from an area 703;

Fig. 11 is an example of the result of area

division of a partial binary image determined as an inverted image by BS processing on the image read from the area 701 in Fig. 8;

Fig. 12 is an example of the result of general
5 BSOCR processing on the inverted image resulted from
pixel inversion processing on the area 701 in Fig. 8;

Fig. 13 is an example of the result of the area division of the partial binary image in Fig. 9;

10 Fig. 14 is an example of the result of the area division of the partial binary image in Fig. 10;

Fig. 15 is an example of an image inverted from the result of the area division of the partial binary image in Fig. 10;

15 Fig. 16 is an example of a second color image subjected to the direction detection processing according to the first embodiment;

Fig. 17 is an example of images resulted from tile header generation with the threshold values $T_1 = 15$ and $T_2 = 1$ from the color image in Fig. 16;

20 Fig. 18 is a graph showing a histogram for base removal for the color image in Fig. 16;

Fig. 19 is an example of processed result showing 3 rectangular areas finally extracted by the primary character extraction processing;

25 Fig. 20 is an example of a read image (partial binary image) from an area 1901;

Fig. 21 is an example of a read image (partial

binary image) from an area 1902;

Fig. 22 is an example of a read image (partial binary image) from an area 1903;

Fig. 23 is an example of the result of the area 5 division of the partial binary image in Fig. 21;

Fig. 24 is an example of the result of the area division of the partial binary image in Fig. 22;

Fig. 25 is a block diagram showing the construction of the image processing apparatus having a 10 multivalue-image reading function, according to a second embodiment of the present invention;

Fig. 26 is a block diagram showing the detailed construction of a primary character extraction unit to perform the primary character extraction processing;

15 Fig. 27 is a flowchart showing the detailed processing procedure by a character rectangle extraction unit 2602 to perform the primary character extraction processing (S310);

Fig. 28 is a flowchart showing the detailed 20 processing procedure by a random access rectangle extraction unit 2603;

Fig. 29A is a functional block diagram showing the construction of a digital color copier according to an embodiment of the present invention;

25 Fig. 29B is a functional block diagram showing the details of a system controller 2150 in a controller unit 2000;

Fig. 29C is a functional block diagram showing the details of a tile image processor 2149 in the controller unit 2000;

Fig. 30 is a block diagram showing the outline of
5 the conventional software image direction detection
processing;

Fig. 31 is an example of the respectively 8-bit
RGB color image 1011;

Fig. 32 is an example of the binary image simply
10 binarized from the color image in Fig. 31;

Fig. 33 is an example of the luminance histogram
of the color image in Fig. 31 and the optimum
binarization point 1041;

Fig. 34 is an example of the binary image from
15 the color image 1011 using the binarization point 1041;

Fig. 35 is an example of the result of area
division in a case where the binary image in Fig. 34 is
subjected to area division by the area division
processing procedure 1014;

20 Fig. 36 is a block diagram showing the hardware
construction of the specialized direction detection
board connected to the main board of the monochrome
digital copier; and

Fig. 37 is a timing chart showing the respective
25 operations using the direction detection board in Fig.
36.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Hereinbelow, operations of image processing method and apparatus according to the present invention will be described in detail with reference to the drawings.

<First Embodiment>

Fig. 1 is a block diagram showing the construction of an image processing apparatus according to a first embodiment of the present invention. In Fig. 1, numeral 101 denotes a color image as the object of direction determination; and 102, a general image processing system having a CPU, a RAM, a ROM and the like and having hardware functions to perform differential processing of input image, histogram operation, compression/decompression processing, resolution conversion, binarization processing and the like. Note that these functions may be performed by software processing in place of hardware.

Numeral 103 denotes differential information generated by the image processing system 102; 104, histogram information (whole histogram) generated by the image processing system 102; and 105, a binary image of character portion (1 area) of color image data generated by the image processing system 102.

The respective parts in a direction detection

module 100 are realized by a processor included in the image processing system 102, however, it may be arranged such that they are realized as software functions, not as hardware functions, and performed
5 using a CPU.

In Fig. 1, numeral 106 in the direction detection module 100 denotes a primary character extraction unit to input the differential information 103 and estimate a character area; 107, a character area coordinates 10 generated by the primary character extraction unit 106; 108, a binarization threshold calculation unit to input the histogram information 104 and calculate an optimum binarization threshold value; and 109, a binarization threshold value calculated by the binarization 15 threshold calculation unit 108.

Further, numeral 110 denotes an area division unit to input the binary image 105 of character portion and generate more detailed coordinates of the character area. The area division unit 110 has an inversion 20 determination function and an inversion processing function. If it is determined that the partial binary image is an inverted area, the area division unit 110 performs inversion processing on the binary image and performs area division processing on the image.
25 Numeral 111 denotes an OCR unit to perform direction determination OCR processing on the character area detected by the area division unit 110.

Next, as an example of the general image processing system 102, a digital color copier will be described with reference to Figs. 29A to 29C.

Fig. 29A is a functional block diagram showing 5 the construction of a digital color copier according to an embodiment of the present invention. In Fig. 1, a controller unit 2000 inputs/outputs image information and device information and performs image mapping of PDL data and the like. The controller unit 2000 is 10 connected to a scanner 2070 as an image input device, a printer 2095 as an image output device, a LAN 2011 and a public line (WAN) 2051.

Fig. 29B is a functional block diagram showing the details of a system controller 2150 in the 15 controller unit 2000. Fig. 29C is a functional block diagram showing the details of a tile image processor 2149 in the controller unit 2000.

In Fig. 29B, a CPU 2001 is a processor to control the overall digital color copier. Note that in the 20 present embodiment, 2 CPUs are connected to a common CPU bus 2126 and further connected to a system bus bridge 2007.

The system bus bridge 2007 is a bus switch connected to the CPU bus 2126, a RAM controller 2124, a 25 ROM controller 2125, an IO bus 2127, a sub bus switch 2128, an IO bus 2129, an image ring interface 2147 and an image ring interface 2148.

The sub bus switch 2128 is a second bus switch connected to an image DMA 2130, an image DMA 2132, a font expansion unit 2134, a sort circuit 2135 and a bitmap trace unit 2136, to arbitrate memory access 5 requests outputted from these DMAs for connection with the system bus bridge.

A RAM 2002 is a system work memory for the CPU 2001 to operate, which also serves as an image memory for temporarily storing image data under the control of 10 the RAM controller 2124. A ROM 2003 is a boot ROM holding a system boot program, which is controlled by the ROM controller 2125.

The image DMA 2130, connected to an image compression unit 2131, controls the image compression 15 unit 21331 based on information set via a register access ring 2137, to read non-compressed data from the ROM 2003, compress the data, and write the compressed data back.

The image DMA 2132, connected to an image expansion unit 2133, control the image expansion unit 2133 based on information set via the register access ring 217, to read compressed data from the RAM 2002, expand the data, and write the expanded data back.

The font expansion unit 2134 expands compressed 25 font data stored in the ROM 2003 or the RAM 2002 based on a font code included in PDL data transferred from the outside via an LAN controller 2010.

The sort circuit 2135 rearranges the order of objects in a display list generated upon bitmapping the PDL data. Further, the bitmap trace circuit 2136 extracts edge information from the bitmap data.

5 The IO bus 2127 is a kind of internal IO bus connected to a USB controller as a standard bus, a USB interface 2138, a general serial port 2139, an interrupt controller 2140 and a GPIO interface 2141. The IO bus 2127 includes a bus arbiter (not shown).

10 An operation interface 2006 is an interface with an operation unit (UI) 2012, which outputs image data to be displayed on the operation unit 2012. Further, the operation interface 2006 transmits information inputted from the operation unit 2012 by a user of the 15 present system to the CPU 2001.

The IO bus 2129 is a kind of internal IO bus connected to a general bus interface 2142 and the LAN controller 2010. The IO bus 2129 includes a bus arbiter (not shown). Further, the general bus 20 interface 2142, comprised of 2 same bus interfaces, serves as a bus bridge to support a standard IO bus. Note that in the present embodiment, PCI buses 2143 are employed.

An external storage unit (HDD) 2004 is a hard disk drive holding system software, image data and the like, connected to one PCI bus 2143 via a disk controller 2144. Further, the LAN controller 2010,

connected to the LAN 2011 via a MAC circuit 2145 and a PHY/PMD circuit 2146 for information input/output. Further, a modem 2050 is connected to the public line 2051 for information input/output.

5 The image ring interface 2147 and the image ring interface 2148 are DMA controllers to connect the system bus bridge 2007 to an image ring 2008 for high-speed image data transfer, so as to transfer compressed tile data between the ROM 2002 and a tile image 10 processor 2149.

The image ring 2008 is comprised of a combination of pair of unidirectional connection paths. The image ring 2008 is connected to a tile expansion unit 2103, a command processor 2104, a status processor 2105 and a tile compression unit 2106 via an image ring interface 2101 and a tile image interface 2102, in the tile image 15 processor 2149. In the present embodiment, 2 tile expansion units 2103 and 3 tile compression units 2106 are employed.

20 The tile expansion unit 2103 is a bus bridge connected to the image ring interface 2101 and a tile bus 2107, to expand compressed image data inputted from the image ring 2008 and transfer the data to the tile bus 2107. The tile compression unit 2106 is a bus bridge connected to the image ring interface 2102 and the tile bus 2107, to compress uncompressed image data 25 inputted from the tile bus and transfer the data to the

image ring 2008.

The command processor 2104, connected to the image ring interface 2101 and the image ring interface 2102 and a register setting bus 2109, to write a 5 register setting request issued from the CPU 2001 and inputted via the image ring 2008 into a corresponding block connected to the register setting bus 2109. Further, based on a register reading request issued from the CPU 2001, the command processor 2104 reads 10 information from a corresponding register via the register setting bus 2109, and transfers the information to the image ring interface 2102.

The status processor 2105 monitors information in the respective image processors, generates an interrupt 15 packet to issue an interrupt to the CPU 2001, and outputs the interrupt packet to the image ring interface 2102. Further, the tile bus 2107 is connected to, in addition to the above blocks, a rendering unit interface 2110, an image input interface 20 2112, an image output interface 2113, a multivalue conversion unit 2119, a binarization unit 2118, a color space conversion unit 2117, an image rotation unit 2030 and a resolution conversion unit 2116.

The rendering unit interface 2110 inputs a bitmap 25 image generated by a rendering unit 2060 to be described later. The rendering unit 2060 and the rendering unit interface 2110 are connected by a

general video signal 2111. The rendering unit interface 2110 makes connection to a memory bus 2108 and the register setting bus 2109 in addition to the tile bus 2107, to convert an input raster image to a tile image 5 by a predetermined method set via the register setting bus 2109 and at the same time performs clock initialization, and to output the tile image to the tile bus 2107.

The image input interface 2112 inputs raster 10 image data subjected to correction image processing by the scanner image processor 2114 to be described later, converts the data to a tile image by a predetermined method set via the register setting bus 2109 and at the same time performs clock synchronization, and outputs 15 the tile image to the tile bus 2107.

The image output interface 2113 inputs the tile image data from the tile bus 2107, converts the data to a raster image and changes a clock rate, and outputs the raster image to a printer image processor 2115.

20 Then image rotation unit 2030 rotates image data. Further, the resolution conversion unit 2116 changes image resolution. The color space conversion unit 2117 converts color space of color and grayscale image. The binarization unit 2118 binarizes a multivalue (color 25 and grayscale) image. The multivalue conversion unit 2119 converts a binary image to multivalue data.

The external bus interface unit 2120 is a bus

bridge to convert and output writing and reading requests, issued from the CPU 2001 via the image ring interfaces 2101, 2102, 2147 and 2148, the command processor 2104 and the register setting bus 2109, to an external bus 2121. In the present embodiment, the external bus 2121 is connected to the printer image processor 2115 and the scanner image processor 2114.

A memory controller 2122, connected to the memory bus 2108, performs writing and reading of image data to/from an image memory 2123 divided into an image memory A and an image memory B by a predetermined address division, in accordance with requests from the respective image processors, and performs a refresh operation or the like in accordance with necessity.

The scanner image processor 2114 performs correction image processing on image data scanned by the scanner 2070 as an image input device. The printer image processor 2115 performs correction image processing for printer output, and outputs the result of processing to the printer 2095.

The rendering unit 2060 converts PDL code or an intermediate display list to a bitmap image. The construction of the digital color copier is as described above.

Next, the flow of the direction determination processing of the present invention will be described about a case where the above-described digital color

copier is employed as the image processing system 102.

Fig. 2 is a block diagram showing a hardware construction necessary for direction detection processing in a case where the digital color copier in 5 Figs. 29A to 29C is applied to the general image processing system 102. As shown in Fig. 2, numeral 201 denotes a color image as the object of direction detection; and 202, a general image processing system having a CPU, a RAM, a ROM and the like and having 10 hardware functions to perform differential processing of input image, histogram operation, compression/decompression processing, resolution conversion, binarization processing and the like, as in the case of the system in Fig. 1. Note that these 15 functions may be performed by software processing in place of hardware.

Further, as in the case of Fig. 1, the respective parts in a direction detection module 200 are realized by a processor included in the image processing system 20, however, it may be arranged such that they are realized as software functions and performed using a CPU (not shown) (CPU 2001 in the above-described 25 digital color copier).

Hereinbelow, generation of a tile header 203, the 25 histogram information (whole histogram) 204 and the 300 dpi partial binary image 205 will be described with reference to Figs. 29A to 29C and Fig. 2.

First, plural color images are placed on an ADF of the scanner 2070. In Fig. 2, the color image 201 indicates each of the plural color images sequentially read by the scanner 2070. Note that the color image is
5 read in 600 dpi.

The scanner image processor 2114 inputs the color image 201 and generates an image area flag 212 as 1-bit information per 1 pixel. That is, the image area flag 212 is generated in 600 dpi. The generation of the
10 image area flag 212 is basically differential processing, i.e., setting a window in the color image 201, applying a differential filter to the image, and setting the flag to "1" if the result of differential processing is over a threshold value while setting the
15 flag "0" if the result of processing is not over the threshold value. Since this processing performed by hardware, it is performed at a high speed, and post processing to eliminate "1" appeared on a halftone natural image is also performed. The image area flag information is stored in 1 bit of a channel for the color image. The degree of JPEG compression by the image processor 2149 and a coefficient for output color processing by the printer image processor 2115 are controlled by this information.

25 The image area flag generation hardware can select a processing signal by setting R, G, B or $a \times R + b \times G + c \times B$ and a coefficient in an internal register.

Generally, as a default signal is a G signal, it is assumed in the present embodiment that the G signal is processed and image area flag is generated.

On the other hand, the scanner image processor 5 2114 performs histogram operation for base level judgment, to generate the whole histogram 204. That is, a histogram of 256 grayscale levels (8 bits) is created for the R, G, B and ND ($=A_1 \times R + A_2 \times G + A_3 \times B$) signals. The histogram is used as data for calculation 10 of parameter for "base removal" performed by the printer image processor 2115.

The image input interface 2112 performs structural conversion to a tile image. That is, the interface generates the tile header 203 using the image 15 area flag information from image area flag distribution in the tile (32 × 32 pixels) (accordingly, the tile header information is a $600/32 =$ about 18 dpi image).

That is, in the image processing method according to the present invention, the tile image is a low 20 resolution binary image, generated by counting binary images generated by differentiating an input image by small area.

As the generation of the image area flag information tile header, "if an image area flag equal 25 to or greater than a threshold value T1 exists in a tile, the tile is set to 1". As the order of image input in the image input interface 2112 is a raster

sequential order, a 10-bit counter for image width/32 is necessary, and the scale of hardware is increased. Accordingly, in this embodiment, based on a rule that "if an image area flag equal to or greater than a 5 threshold value T1 exists in even 1 of tile 32 lines, the tile is set to 1", the image area flag information tile header is generated.

That is, in the image processing method according to the present invention, a binary image of image area 10 flags having a value "1" if a corresponding pixel is equal to or greater than a predetermined value, or a value "0" if a corresponding pixel is less than the predetermined value, is generated, and a tile image is generated such that in a tile header, a tile where the 15 number of "1" image area flags is equal to greater than a predetermined threshold value is set to "1", and a tile where the number of "1" image area flags is less than the predetermined threshold value is set to "0".

Further, in this embodiment, the image area flag 20 information tile header is generated for 2 bits. That is, 1 bit is generated as "if an image area flag equal to or greater than a threshold value T1 exists in even 1 of tile 32 lines, the tile is set to 1", and the other 1 bit is generated as "if an image area flag 25 equal to or greater than a threshold value T2 exists in even 1 of tile 32 lines, the tile is set to 1" ($T_1 > T_2$).

That is, in the image processing method according to the present invention, plural tile images are generated using plural different threshold values, then a comparison is made among the plural tile images, and

5 a character area included in the input image is extracted. Further, the feature of the image processing method according to the present invention is that plural low resolution images are generated using plural different threshold values. Further feature of

10 the image processing method according to the present invention is that a group of connected pixels extracted from plural low resolution images is compared with the plural low resolution images, and a character area is extracted.

15 Note that as the image area flag 212 is differential information, a large number of flags 212 appear in a portion of a differential-filtered area with a large level difference. Accordingly, in case of a black character on white base, a large number of

20 image area flags 212 appear, however, in case of a character with a small level difference between the character and the base, the number of flags is smaller than that of the black character. By utilizing this characteristic, 2 image area flag distribution

25 information (tile headers) 203 are generated from the threshold values T1 and T2, and further accurate character area coordinate detection is performed by the

primary character extraction unit 206.

That is, in the image processing method according to the present invention, the tile image is a low resolution differential image generated by counting differential information of input image by small area.

The color image data (RGB), α channel data including the image area flag data and the tile header information are compressed in the image processor 2149. More specifically, the RGB data is subjected to JPEG compression and the α channel data is subjected to reversible compression in the tile compression unit 2106 (not shown in Fig. 2) in the image processor 2149. As the tile header itself is not compressed, the tile header information can be read at a high speed. Then the data is spooled in the RAM 2002.

Next, a procedure for generating the 300 dpi partial binary image 205 from the spooled image will be described.

The image spooled in the RAM 2002 is delivered through the image ring interfaces 2147 and 2101 to the image processor 2149. The image data expanded through the tile expansion unit 2103 is converted from 600 dpi to 300 dpi in the resolution conversion unit 2116. Thereafter, the data is binarized (to 00 and ff values) with a designated threshold value in the color space conversion unit 2117. The result of 8 to 1 bit conversion of the data is returned through the image

ring interfaces 2102 and 2148 to the system controller 2150. Then, a partial binary image is written into the RAM 2002.

As described above, in the image processing system (color digital copier), the tile header 203, the hole histogram 204, and the 300 dpi partial binary image 205 are generated.

Fig. 3 is a timing chart showing respective operations of the image processing system 202 in Fig. 2. Hereinbelow, the flow of processing in the image processing system 202 will be described with reference to Fig. 3. In Fig. 3, numeral 301 denotes the timing of the scanner and respective numerical values "0" to "4" indicates the order of color images (originals) placed on the ADF. Further, the timing of the scanner image processor 2114, denoted with numeral 302, is performed in synchronization with the scanner. Note that as the generation of the tile header and the histogram operation for base removal are completed at timing indicated with each symbol " Δ ", the tile header 203 and the whole histogram 204 are formed at this timing.

Further, numeral 303 denotes the timing of direction determination processing which is executed in software processing utilizing the CPU 2001. As the CPU 2001 performs plural tasks, the CPU is not occupied as indicated in timing 306 in Fig. 3 but the other tasks

are performed. However, in the present embodiment, the other tasks are not shown for the sake of convenience. The direction determination processing is started at the timing indicated with each symbol "Δ". In Fig. 3, numerals 305 and 306 denote the details of the direction determination processing on some color image. Note that numeral 305 denotes hardware processing; and 306, software processing.

The software processing 306 will be described in detail. First, the histogram data is read (step S307). More particularly, the CPU refers to the histogram data in the scanner image processor. The histogram to be read in this case is a G histogram corresponding to the signal used in the image area flag generation. The CPU performs the binarization threshold calculation software to obtain a threshold value A (step S308). Next, the CPU reads the tile header existing in the RAM (step S309). After the reading of the tile header, the CPU performs the primary character extraction processing (step S310).

Note that the primary character extraction processing means limiting a portion including characters and selecting the portion as a small rectangular area from the entire image. Fig. 26 is a block diagram showing the detailed construction of the primary character extraction unit to perform the primary character extraction processing by hardware.

Note that in the present embodiment, the primary character extraction processing is executed as software processing.

In Fig. 26, a tile header input unit 2601 inputs 5 2 tile header images 203 spooled in the RAM 2002. Further, a character rectangle extraction unit 2602 extracts a rectangular area with a high probability of existence of a set of characters, from the 2 tile header images. Further, a random access rectangle 10 extraction unit 2603 simply divides the entire image into approximately uniform rectangular areas, and extracts a rectangular area with at least a low probability of existence of character. Then, a character area output unit 2604 integrates the group of 15 rectangles extracted by the character rectangle extraction unit 2602 and the random access rectangle extraction unit 2603, and outputs as plural character areas.

Fig. 27 is a flowchart showing the detailed 20 processing procedure by the character rectangle extraction unit 2602 to perform the primary character extraction processing (S310). First, all the connected black pixels are extracted from tile header 1 image, and coordinates of respective circumscribed rectangles 25 thereof are obtained (step S2701). The set of rectangular information is referred to as {Tc1}.

Next, an average size of all the rectangles is

obtained except rectangles having an extremely large/small area and a high oblateness among the set of rectangles {Tc1} (step S2702). Further, a set of rectangles having a size greatly different from the 5 average size of the set of rectangles {Tc1} is determined as a non-text rectangles {Tc1-nt}, and others, as text rectangles {Tc1-t} (step S2703).

Further, all the connected black pixels are extracted from the tile header 2 image, and coordinates 10 of the respective circumscribed rectangles are obtained (step S2704). The set of rectangular information is referred to as {Tc2}. Note that in the present embodiment, at steps S2701 and S2704, the processing of extracting the connected black pixels is performed 15 using well-known labeling, however, it may be performed using other well-known edge line tracing.

That is, the feature of the image processing method according to the present invention is extracting an area in the image, corresponding to connected pixels 20 extracted from a low-resolution image, as a character area.

Further, in the set of rectangles {Tc2}, a set of rectangles having an area equal to or smaller than a predetermined area is determined as a set of non-text 25 rectangles {Tc2-nt}, and a set of rectangles having an area larger than the predetermined area, as text rectangles {Tc1-t} (step S2705). The threshold value

of determination by area is a previously-determined appropriate value. Then the coordinates of the respective rectangles are converted from 18 dpi tile header coordinate values to 300 dpi coordinate values

5 (step S2706).

Further, regarding each rectangle of the set of rectangles {Tc2-t}, the total sum of areas among the set of rectangles {Tc1-t} having coordinates overlapped with the rectangle is divided by the area of the

10 rectangle, and the obtained value is set as a score of the rectangle (Step S2707). A rectangle having a high score is considered as an area with a high probability of characters observed as text in the both tile header 1 and the tile header 2.

15 The set of rectangles {Tc2-t} are sorted in ascending order of the scores (step S2708). The sorted set of rectangles {Tc2-1} becomes the result of processing by the character rectangle extraction unit 2602.

20 On the other hand, Fig. 28 is a flowchart showing the detailed processing procedure by the random access rectangle extraction unit 2603. First, the tile header image 2 is divided into meshes by a small rectangle having a predetermined area (step S2801). Note that

25 the mesh area is previously determined in consideration of a partial image size for efficient processing in the subsequent stage. However, the mesh area may be

dynamically changed in correspondence with the size of input image. Further, to avoid noise which occurs in an image peripheral portion, a margin may be set except a predetermined peripheral width before mesh division
5 is performed.

Next, in each mesh, the total sum S_m of areas of the $\{T_{c2}\}$ rectangles included in the mesh is obtained (step S2802). Then 4 adjacent meshes are defined as 1 random access rectangle T_r (step S2803). Then the
10 total sums S_m of the 4 meshes is added up as a score of the T_r rectangle.

Next, the set of all the T_r rectangles, $\{T_r\}$ is sorted in ascending order of the scores (step S2804). At this time, overlap between T_r rectangles is avoided.
15 If overlapped 2 T_r rectangles exist, an overlapped portion of a lower score rectangle is removed and registered again as a new T_r . Then the T_r rectangles are sorted again in ascending order of the scores. The sorted set of rectangles $\{T_r\}$ becomes the result of
20 processing by the random access rectangle extraction unit 2603.

Finally, the output unit 2604 outputs rectangles having a score equal to or greater than a predetermined threshold value in descending order, from the set of
25 rectangles $\{T_{c2-1}\}$. The output unit abandons remaining T_{c2-t} rectangles having scores less than the threshold value, and outputs the set of rectangles $\{T_r\}$ in the

order of scores. When there is no rectangle or a predetermined number of rectangles have been outputted, the processing in the output unit 2604 ends.

That is, the feature of the image processing method according to the present invention is that a low resolution image is divided into meshes, and a character area is extracted based on distribution of pixels within each mesh area. Further, another feature of the image processing method according to the present invention is that a character area extracted using connected pixels extracted from a low resolution image and a character area determined based on distribution of pixels within each mesh area are selectively outputted.

Plural character area coordinates are obtained by the above-described primary character extraction. An image reading table is generated on the RAM in accordance with the result of processing, and when the CPU makes a program kick, the 300 dpi partial binary images 205 are sequentially read by DMA processing by the image ring interface 2147. Upon completion of reading from the first area (area 1) (step S311), area division processing (hereinbelow, referred to as "BS processing") and the OCR task are performed on the area 1 (step S319) (hereinbelow, referred to as "BSOCR processing"), and the result of processing is stored in the BSOCR task.

In parallel to the BSOCR processing on the area 1, the 300 dpi binary image from an area 2 is performed. As the BSOCR processing requires sufficient time in comparison with image reading (hardware processing),
5 the image data is provided prior to the BSOCR processing. The result of BSOCR processing on the area 2 is stored and added in the BSOCR task. When the processing time runs out, the BSOCR task calculates a final result of direction determination from the result
10 of stored results and outputs the final result.

Next, a particular example of the above-described direction determination processing will be described with reference to actual 2 images. In this case, a color image as shown in Fig. 4 is processed. Fig. 4 shows an
15 example of a color image used in the direction determination processing according to the first embodiment. Fig. 5 shows an image as a result of tile header generation with threshold values $T_1 = 15$ and $T_2 = 1$ in the scanner image processor 2114. In Fig. 5, an
20 image 501 is a result of processing with the threshold $T_1 = 15$, and an image 502, a result of processing with the threshold $T_2 = 1$. On the other hand, in the scanner image processor 2114, the histogram operation for base removal is not performed but a histogram as
25 shown in Fig. 6 is obtained. Fig. 6 shows a whole histogram of the color image in Fig. 4. Then, an optimum binarization threshold value $A = 128$ denoted

with numeral 601 is calculated from the histogram in Fig. 6.

The primary character extraction unit outputs rectangular areas arranged in the order of scores of probability of character area by the processing procedure shown in the flowchart of Fig. 27. Fig. 7 is an example of a processed result showing 3 rectangular areas finally extracted by the primary character extraction processing. Note that actually more areas are extracted from this image, however, for the sake of convenience, 3 areas are extracted. Further, as rectangles obtained by the random access rectangle extraction are positioned in lower ranks in the score order, they are ignored.

In Fig. 7, an image reading table for areas 701 to 703 is generated on the RAM, and when the image ring interface is kicked, resolution conversion are performed on the respective areas, and binary images are sequentially read. Fig. 8 shows a read image (partial binary image) from the area 701. Fig. 9 shows a read image (partial binary image) from the area 702. Fig. 10 shows a read image (partial binary image) from the area 703.

Upon completion of reading from the area 701, the direction determination enters the BSOCR processing loop. The BSOCR processing for the area 701 is performed by the CPU. The BS (area division)

processing includes image inversion determination processing and inversion processing. More particularly, the number of black pixels and the number of white pixels of an input binary image are counted, and if the 5 percentage of black pixels is e.g. 70% , it is determined that the image is an inverted image.

Fig. 11 shows the result of area division of a partial binary image determined as an inverted image by the BS processing on the image read from the area 701 10 in Fig. 8. Fig. 12 is an example of the result of general BSOCR processing on the inverted image resulted from pixel inversion processing on the area 701 in Fig. 8. In Fig. 11, an area indicated with a red rectangular RD1 is recognized as a character area. The 15 display of rectangle is made as white characters on black base according to the specifications of a GUI application. An actual image is inverted, and the image in Fig. 12 is subjected to the BS processing and the area resulted from the BS processing surrounded 20 with a black rectangle BK1 is subjected to the OCR processing.

Note that it is preferable that in the OCR processing, the result of direction determination is the total of processing results in plural areas. For 25 example, in a case where 200 characters exist in 1 area, to avoid time out of processing time while 4 direction OCR processing is performed on the 200 characters, the

number of characters per 1 area is limited. That is,
the limited number of characters per 1 area is 30. As
a result of OCR processing of the image in Fig. 12, the
accuracy of 0° in processing of 30 characters is
5 increased. Further, if processing time remains, the
BSOCR processing is performed on the previously-read
image of the area 702. The area 702 is the binary
image in Fig. 9. It is determined by the BS processing
that this area is not an inverted image, and it is
10 determined that an area indicated with a black
rectangle RD2 in Fig. 13 is character area. Fig. 13
shows the result of area division of the partial binary
image in Fig. 9. This area is subjected to the OCR
processing, and further the result of direction
15 determination of 30 characters is obtained.

When the areas 701 and 702 have been processed,
if processing time remains, the BSOCR processing is
performed on the previously-read image of the area 703.
The area 703 is the binary image in Fig. 10. It is
20 determined by the BS processing that the area is an
inverted image, and pixel inversion processing is
performed as shown in Fig. 15, and an area indicated
with a black rectangle of the binary image in Fig. 15
(red rectangle RD3 in Fig. 14) is determined as a
25 character area and OCR processed. Fig. 14 shows the
result of the area division of the partial binary image
in Fig. 10. Fig. 15 shows an image inverted from the

result of the area division of the partial binary image in Fig. 10. If the processing time expires at this time, the OCR calculates the results of direction recognition stored inside, and outputs the result of 0°.

5 That is, the feature of the image processing method according to the present invention is that it is determined whether or not a character area is an inverted image based on a binary image of an input image, and if it is determined that the character area
10 is an inverted image, black and white components of the binary image are inverted.

Fig. 16 is an example of a color image different from the color image in Fig. 4 subjected to the direction detection processing according to the first
15 embodiment. The level difference of G plane is not large in the entire image, and if the tile header generation is performed using the same threshold values (T1 = 15 and T2 = 1) as those in the processing of the color image in Fig. 4, the results as shown in Fig. 17
20 are obtained. Fig. 17 shows images resulted from the tile header generation with the threshold values T1 = 15 and T2 = 1 from the color image in Fig. 16. On the other hand, Fig. 18 is a graph showing a histogram for base removal for the color image in Fig. 16. As
25 direction determination preprocessing, an optimum binarization threshold value A = 66 denoted with numeral 1801 is calculated from the histogram in Fig.

18.

In the primary character extraction, the processing shown in the flowchart of Fig. 27 is performed on images 1701 and 1702, but appropriate 5 character rectangles cannot be extracted. However, by performing the processing shown in the flowchart of Fig. 28, portions determined as areas including some information which can be or cannot be characters, are outputted as random access rectangles. In this example, 10 3 areas as shown in Fig. 19 are outputted. That is, Fig. 19 shows processed 3 rectangular areas finally extracted by the primary character extraction processing. Note that actually more areas are extracted from this image, however, for the sake of 15 convenience, 3 areas are extracted.

An image reading tables for areas 1901 to 1903 is generated on the RAM and when the image ring interface is kicked, resolution conversion is performed on the respective areas, and binary images are sequentially 20 read. Fig. 20 shows a read image (partial binary image) from the area 1901. Fig. 21 is an example of a read image (partial binary image) from the area 1902. Fig. 22 shows a read image (partial binary image) from an area 1903.

25 Upon completion of reading from the area 1901, the direction determination enters the BSOCR processing loop. The BSOCR processing for the area 1901 is

performed by the CPU. In the image as shown in Fig. 20,
it is determined as the result of area division that
there is no character and the OCR processing is not
performed. In the binary image in Fig. 21, an area
5 surrounded with a red rectangle RD4 in Fig. 23 is
recognized as a character area. That is, Fig. 23 shows
the result of area division on the partial binary image
in Fig. 21. This area is subjected to the OCR
processing and the result of direction determination
10 for 30 characters can be obtained.

When the areas 1901 and 1902 have been processed,
if time remains, the BSOCR processing is performed on
the previously-read image of the area 1903. In the
binary image in Fig. 22, an area surrounded with a red
15 rectangle RD5 in Fig. 24 is recognized by the BS
processing as a character area. That is, Fig. 24 shows
the result of area division of the partial binary image
in Fig. 22. This area is subjected to the OCR
processing, and further result of direction
20 determination for 30 characters can be obtained.

Assuming that the processing time runs out, the OCR
calculates the direction determination results stored
inside and outputs the result of 0°.

As described above, the feature of the image
25 processing method according to the present invention is
that the direction of an image including a character
area inputted into a computer is detected by generating

a binary image, then generating a tile image by adding a predetermined value to tiles each defined as a predetermined size area in the binary image, then extracting an area corresponding to an area within a circumscribed rectangle surrounding connected pixels having the same value in the tile image, as character area, from the binary image, recognizing the directions of the characters included in the character area, and detecting the direction of the image.

Further, the feature of the image processing method according to the present invention is that respective characters included in a character area are extracted, and the direction of the extracted characters is recognized, and the direction of the character area is detected based on the result of recognition of the direction of the characters included in the character area.

<Second Embodiment>

In the above-described first embodiment, the binarization threshold value A is calculated from the whole histogram provided by the image processing system, however, in the present invention, the binarization threshold value may be a fixed value of, e.g., "128".

In case of fixed threshold value, the binarization can be performed even if the image processing system side lacks the whole histogram operation unit.

Further, in the above-described first embodiment, the area division processing is performed on the 300 dpi partial image, however, the processing may be omitted. In this case, the time necessary for the area 5 division processing can be saved, and the OCR processing can be performed on more areas. However, in a case where character cutting cannot be made without any processing from images detected by the primary character extraction processing, the result of 10 direction detection becomes poor. Then the primary character extraction processing may be arranged such that an area to be subjected to the subsequent area division processing and an area not to be subjected to the area division processing are designated. In this 15 case, the accuracy of direction detection can be maintained and the processing speed is increased.

Further, in the first embodiment, the partial image is a 300 dpi binary image, however, the present invention is not limited to this image. It may be 20 arranged such that a partial multivalue image is read. In this case, the direction detection side requires a function of binarizing the partial multivalue image. In this arrangement, optimum binarization can be performed on the partial area.

25 Further, considering data size, the time for reading a multivalue image is longer than for reading a binary image. Accordingly, it is effective that in the

primary character extraction processing, it is determined whether a binary image reading or multivalue image reading is to be performed based on the distribution of differential information, and the image processing system is instructed to select binary image reading or multivalue image reading in accordance with the determination. Fig. 25 is a block diagram showing the construction of the image processing apparatus having a multivalue-image reading function, according to the second embodiment of the present invention. The character area coordinate information includes the designation of reading method (binary or multivalue), and the image processing system reads a partial image by DMA operation in accordance with the designation. In a case where a binary image is read, the same processing as that in the first embodiment is performed. In a case where a multivalue image 2512 is read, partial-area optimum binarization processing 2513 is performed. In a case where this optimum binarization processing includes inversion determination and an inversion function, the inversion determination is not necessary in the subsequent area division processing 2514.

25 <Other Embodiment>

The present invention can be applied to a system constituted by a plurality of devices (e.g., a host computer, an interface, a reader and a printer) or to

an apparatus comprising a single device (e.g., a copy machine or a facsimile apparatus).

Further, the object of the present invention can also be achieved by providing a storage medium (or recording medium) holding software program code for performing the aforesaid processes to a system or an apparatus, reading the program code with a computer (e.g., CPU, MPU) of the system or apparatus from the storage medium, then executing the program. In this case, the program code read from the storage medium realizes the functions according to the embodiments, and the storage medium holding the program code constitutes the invention. Furthermore, besides aforesaid functions according to the above embodiments are realized by executing the program code which is read by a computer, the present invention includes a case where an OS (operating system) or the like working on the computer performs a part or entire actual processing in accordance with designations of the program code and realizes functions according to the above embodiments.

Furthermore, the present invention also includes a case where, after the program code read from the storage medium is written in a function expansion card which is inserted into the computer or in a memory provided in a function expansion unit which is connected to the computer, CPU or the like contained in

the function expansion card or unit performs a part or entire process in accordance with designations of the program code and realizes functions of the above embodiments.

5 In a case where the present invention is applied to the aforesaid storage medium, the storage medium stores program code corresponding to the flowcharts described in the embodiments.

As described above, according to the present
10 invention, as a character area is detected utilizing plural low resolution images comprised of differential information of a color image, an image input direction can be efficiently detected, from an image with much differential information to an image with little
15 differential information. Further, according to the present invention, as the system has the inversion determination unit, the character recognition can be performed in an inverted character portion, and the input direction can be detected even in a color image
20 having many inverted character portions. Further, according to the present invention, as high-speed software processing can be performed with a small work memory, the cost of parts necessary for version updating of the direction detection processing can be
25 reduced.

The present invention is not limited to the above embodiments and various changes and modifications can

be made within the spirit and scope of the present invention. Therefore, to appraise the public of the scope of the present invention, the following claims are made.